

EE 440 Homework #4

(Luke Jiang, 1560831)

1)

In order to generate a noisy image that 15% of the pixels in the image are noisy pixels, I used `rand()` function, which returns a double between 0 and 1. If `rand()` returns a value smaller than 0.15, then this pixel is selected as noise. I then used `randi([0, 1])` to randomly select the noise to be white or black, each case has a 50% probability.

In order to reduce the noise, I first used a 3*3 Low Pass Filter with coefficients equal to 1. I convolved the LPF with the noisy image. I then applied a 3*3 Median Filter to the noisy image. The result of LPF looks blurry and the noisy pixels look less obvious than the noisy version. The MF does not have the blurring effect and removes almost all noise while preserves the details of the original image.

original image



noisy image



apply Low Pass Filter to noisy img

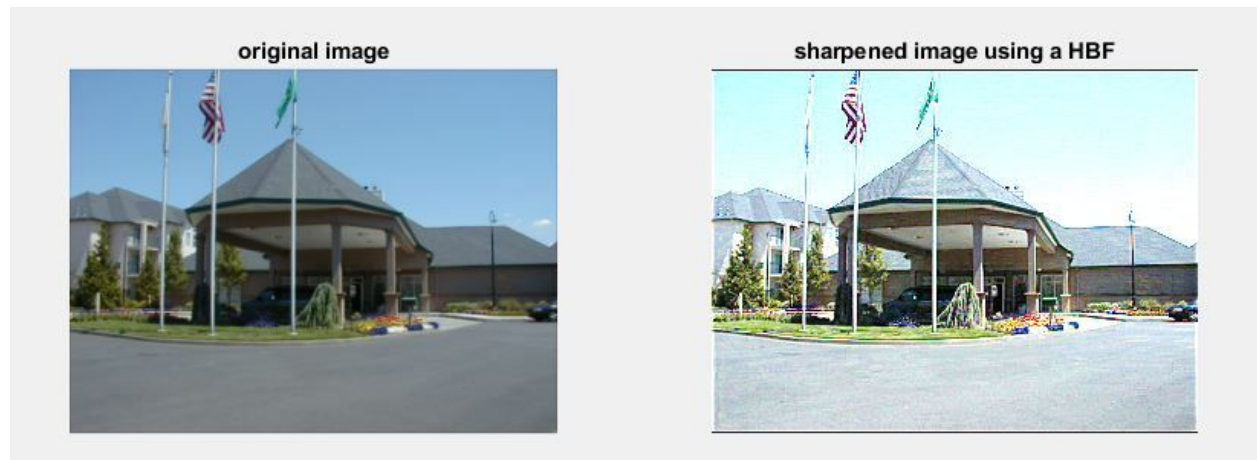


apply Median Filter to noisy img



2)

The high boost filter I used has a scale factor $A = 1.7$



Source Code:

-Part 1:

```

1 - clear all; close all;
2
3 - % -read the original image and display it
4 - img = imread('4_1.bmp');
5 - figure(1); subplot(2, 2, 1);
6 -     imshow(img);
7 -     title('original image');
8
9 - % -generate the noised image
10 - img_n = img;
11 - for i = 1:512
12 -     for j = 1:512
13 -         if (rand() < 0.15) % 15% probability of chosen as noise pixel
14 -             black = randi([0, 1]); % 50% probability of dark/white noise
15 -             if (black) img_n(i,j,:) = [255, 255, 255];
16 -             else img_n(i,j,:) = [0, 0, 0];
17 -             end
18 -         end
19 -     end
20 - end
21
22 - % -display the noisy image
23 - figure(1); subplot(2, 2, 2);
24 -     imshow(img_n);
25 -     title('noisy image');
26
27 - % -using a LPF to reduce noise and display the result
28 - LPF = (1/9) * ones(3, 3);
29 - img_LPF = uint8(zeros(514, 514, 3));
30 - img_LPF(:, :, 1) = conv2(img_n(:, :, 1), LPF);
31 - img_LPF(:, :, 2) = conv2(img_n(:, :, 2), LPF);
32 - img_LPF(:, :, 3) = conv2(img_n(:, :, 3), LPF);
33 - figure(1); subplot(2, 2, 3);
34 -     imshow(img_LPF);
35 -     title('apply Low Pass Filter to noisy img');
36
37 - % -using a MF to reduce noise and display the result
38 - img_n1 = img_n(:, :, 3);
39 - img_MF = uint8(zeros(512, 512, 3));
40 - A_ = uint8(zeros(3, 3)); % preallocate space(3x3 window) for calculating median
41 - for i = 1:510
42 -     for j = 1:510
43 -         A_ = img_n1(i:i+2, j:j+2);
44 -         img_MF(i,j,1) = median(A_(:));
45 -     end
46 - end
47 - img_MF(:, :, 2) = img_MF(:, :, 1);
48 - img_MF(:, :, 3) = img_MF(:, :, 1);
49 - figure(1); subplot(2, 2, 4);
50 -     imshow(img_MF);
51 -     title('apply Median Filter to noisy img');
52

```

-Part 2:

```
1 - clear all; close all;
2
3 - % -read and display the original image
4 - img = imread('4_2.bmp');
5 - figure; subplot(1, 2, 1);
6 -     imshow(img);
7 -     title('original image');
8 - % -create a high boost filter
9 - HBF = [-1 -1 -1; -1 9.7 -1; -1 -1 -1];
10 - % -convolve the HBF with the image to produce the result
11 - res = uint8(zeros(482, 642, 3));
12 - res(:,:,1) = conv2(img(:,:,1), HBF);
13 - res(:,:,2) = conv2(img(:,:,2), HBF);
14 - res(:,:,3) = conv2(img(:,:,3), HBF);
15 - % -display the result
16 - subplot(1, 2, 2);
17 -     imshow(res);
18 -     title('sharpened image using a HBF');
```